

CPSC 1155 – Lab 2

Elementary Programming

Learning Objectives

At the end of this lab, you should be able to:

- Understand and analyze a problem statement
- Write a pseudocode to solve a problem
- Write and run a C++ program to solve a problem

Lab Introduction

This lab provides you with an understanding of elementary programming by understanding problems, writing pseudocodes, and writing C++ programs.

Lab Instructions

Write your answers to the **Practice Questions** in a text editor.

For each **Problem Statement**, follow the steps below:

1. Read the problem statement and clarify the problem.
2. Determine input, process, and output (IPO).
3. Work out the problem by hand using typical values.
4. Declare the variables and constants (data type + meaningful names).
5. Write a pseudocode as required.
6. Write a C++ program (use the given filename) that implements the pseudocode (or algorithm). Add comments where needed.
7. Make sure to use a comments header to reflect the intention of your program and name of the author (you) and the date the program was written.
8. Test, debug, and execute the program using typical values.

Note that sample runs are just examples. Your program should produce correct output for any valid input value.

Submit according to the instruction in the "Lab Submission" section.

Practice Questions

1. [4] **Declaring Variables.** Write C++ declarations (datatype variableName) for storing the following quantities. Choose between integers and floating-point numbers. Declare constants when appropriate. Note that you have to assign values to constants when you declare them.
 - a. The number of days per week
 - b. The number of days until the end of the semester
 - c. The number of centimeters in an inch
 - d. The height of the tallest person in your class, in centimeters

2. [6] **Convert Algorithm to Pseudocode and C++ Program.** The following algorithm is supposed to read a value for miles, convert it to kilometer and print the results. Translate the algorithm first into a pseudocode and then to a C++ program. Test your program in a code editor. Write the pseudocode and C++ program as your answer in the text file.

Step 1: Declare a double variable named miles.

Step 2: Get a value from the user for miles.

Step 3: Declare a double constant named KILOMETERS_PER_MILE with value 1.609.

Step 4: Declare a double variable named kilometers, multiply miles and KILOMETERS_PER_MILE, and assign the result to kilometers.

Step 5: Display kilometers to the console along with an informative message.

3. [4] **Reorder a Program.** The following C++ program is supposed to compute the area of a wall with two windows, but the statements are not in the correct order. Rearrange the statement to correctly calculate the area. Make sure the comments correctly describe the following lines. Write and test your program in a code editor (remember to add the libraries). Write the correct order of statements by writing the line numbers as your answer in the text file.

Line #	Statement
1	double wall_height = 8;
2	int windows = 2;
3	int main()
4	double window_width = 3;
5	Dimensions of the wall and
6	number and dimensions of the windows
7	cout << "Area of wall with 2 windows is: " << area;
8	/*
9	double window_height = 4;
10	area = area - windows * window_width * window_height;
11	*/
12	double wall_width = 30;
13	*/
14	/*
15	This program computes the area of a wall with two windows.
16	// Computation and display of the result
17	double area = wall _width * wall_height;
18	{
19	}

4. [10] **Problem Solving.** Write your answers for each step in the text file:

Problem Statement You are asked to calculate the amount a customer owes. A customer buys two items. One item is 20% off and the other one is at regular price. The amount of tax is 12% for all items.

Step 1 Understand the problem: What are the inputs? What are the desired outputs? What are the constants?

Step 2 Work out examples by hand. Assume the prices for the items and calculate the total amount the customer owes. Think of all the possible values (valid or invalid) for the input.

Step 3 Declare the variables and constants that you need with meaningful names and correct data types. Note that you need a variable for the discount (20%) because it may change later. What is the range of valid values for the input?

Step 4 Write a pseudocode that reads the input values (clearly ask the user to input valid values), calculates the results, and displays the results with an informative message.

Step 5 Turn the pseudocode into a C++ program. Add comments where needed. Run and test your program with typical values.

Note that your program easily works for different values of tax and discount. You only need to change the values where you declared the variables.

Problem Statements

5. [6] (feet_to_meter.cpp) Write a **pseudocode** and **C++ program** that reads a number in feet, converts it to meters, and displays the result. One foot is .305 meter. Here is a sample run:

```
Enter a value for feet: 16.5
16.5 feet is 5.0325 meters
```

6. [4] (celsius_to_fahrenheit.cpp) Write a **C++ program** that reads a Celsius degree, converts it to Fahrenheit, and displays the result. Be careful about the data types.

$$fahrenheit = \frac{9 \cdot celsius}{5} + 32$$

Here is a sample run:

```
Enter a degree in Celsius: 43
43 in Celsius is 109.4 Fahrenheit
```

7. [6] (rectangle.cpp) Write a **pseudocode** and **C++ program** that reads in the width and height of a rectangle, calculates and displays the area and perimeter of the rectangle. The input should be of type double.

Here is a sample run:

```
Enter the width and height of a rectangle: 12 5.2
The area is 62.4
The perimeter is 34.4
```

Lab Submission

Submit a zip folder named as yourName_Lab2.zip to Brightspace. This folder should consist of a **text file** named answer.txt with your answers to Practice Questions, **C++ codes** in individual .cpp files, and one pseudocode.txt file with all your **pseudocodes** to the Problem Statements. Only submit pseudocode for the questions that require it. Review the Coding Guideline on Brightspace for example folder structure.

Please make sure that all your .cpp files compile and run properly before submission. Your file must run properly in order to receive full marks.

Marking Scheme

The marks are given in square brackets [] for each question.